

Connecting with Go

This guide walks you through setting up a Go application to connect to a TimescaleDB database, using the PostgreSQL-compatible `lib/pq` driver, and running a basic query to verify the connection.

Variables


To connect to a TimescaleDB database, you only need **one environment variable** — the connection URI. This URI contains all the necessary information like username, password, host, port, and database name.

Variable	Description	Purpose
<code>TIMESCALE_URI</code>	Full TimescaleDB (PostgreSQL-compatible) connection string from the Elestio service overview	Provides all credentials and connection details in a single URI

A typical URI format looks like:

```
postgresql://<USER>:<PASSWORD>@<HOST>:<PORT>/<DATABASE>
```

You can find the details needed in the URI from the **Elestio service overview** details. Copy and replace the variables carefully in the URI example provided above.


 **timescaledb-gi7jy**

TimescaleDB

Cluster

Running

>_ Open terminal


 Delete cluster

Add node

OverviewNodesBackupsAudit


Termination protection

Disabled. VM can be powered off and terminated.

Protection deactivated 

Auto-Failover

Enabled. In case of failure, the cluster will automatically attempt to recover

Auto-Failover activated 






Node

1 Primary Node

Database Admin

Display your database credentials

Hide DB Credentials

Host	timescaledb-gi7jy-u7774.vm.elestialio.app	
Port	25432	
User	postgres	
Password	*****	Show password 
CLI	PGPASSWORD=***** psql --host=timescaledb-gi7jy-u7774.vm.elestialio.app --port=25432 --username=postgres	Show password 

Prerequisites

• Install Go

- Check if Go is installed:

```
go version
```

- If not, download and install Go: <https://go.dev/dl/>

• Install pq Driver

```
go get github.com/lib/pq
```

Code

Once all prerequisites are set up, create a new file named `main.go` and add the following code, and replace the `TIMESCALE_URI` with actual link or in environment setup as you wish:

```
package main
```

```

import (
    "database/sql"
    "fmt"
    "log"
    "os"

    _ "github.com/lib/pq"
)

func getDBConnection(connStr string) (*sql.DB, error) {
    db, err := sql.Open("postgres", connStr)
    if err != nil {
        return nil, fmt.Errorf("failed to open database connection: %v", err)
    }

    if err := db.Ping(); err != nil {
        return nil, fmt.Errorf("failed to ping database: %v", err)
    }

    return db, nil
}

func main() {
    // Get the TimescaleDB connection string from environment variable
    connStr := os.Getenv("TIMESCALE_URI")
    if connStr == "" {
        log.Fatal("TIMESCALE_URI environment variable not set")
    }

    db, err := getDBConnection(connStr)
    if err != nil {
        log.Fatal(err)
    }
    defer db.Close()

    query := "SELECT current_database(), current_user, version()"
    row := db.QueryRow(query)

    var dbName, user, version string
    if err := row.Scan(&dbName, &user, &version); err != nil {

```

```
    log.Fatal("Failed to scan row:", err)
  }

  fmt.Printf("Connected to TimescaleDB\nDatabase: %s\nUser: %s\nVersion: %s\n", dbName, user,
version)
}
```

Set your TimescaleDB URI as an environment variable:

```
export TIMESCALE_URI=postgresql://user:password@host:port/database
```

To execute the script, open the terminal or command prompt and navigate to the directory where `main.go`. Once in the correct directory, run the script with the command

```
go run main.go
```

If successful, you'll see output like:

```
Connected to TimescaleDB
Database: elestio
User: postgres
Version: PostgreSQL 14.13 (Debian 14.13-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc
(Debian 12.2.0-14) 12.2.0, 64-bit
```

Revision #1

Created 13 May 2025 07:33:00 by kaiwalya

Updated 13 May 2025 07:34:34 by kaiwalya